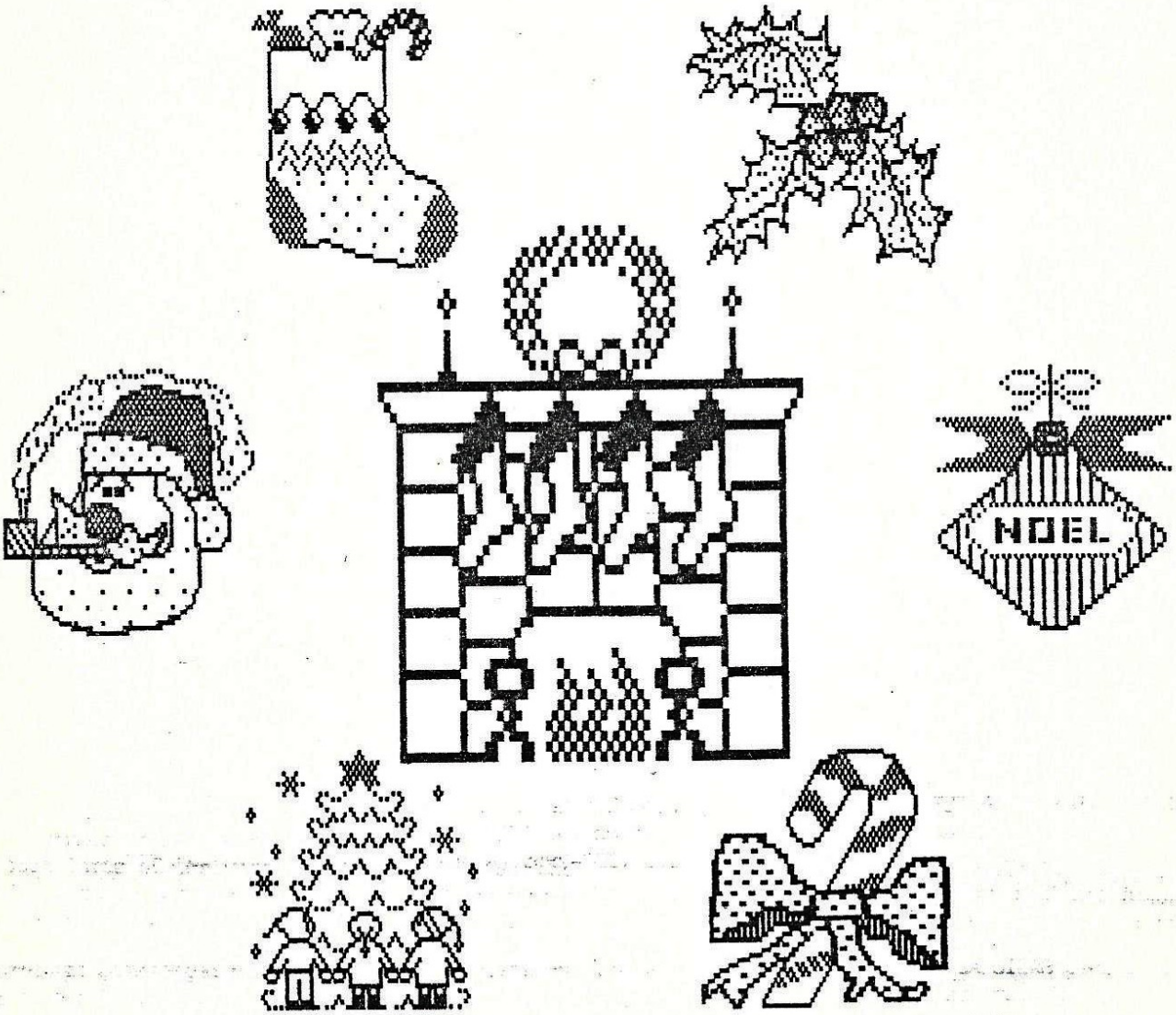
Wire TAPS

THE NEWSLETTER OF THE ATARI PROGRAMMERS' SOCIETY

DECEMBER 1985 VOLUME 3 NUMBER 12



W i r e T A P S -- The Newsletter of THE Atari Programmers' Society

WireTAPS is published almost monthly by The Atari Programmers' Society (TAPS), Toronto, Ontario. All articles in this newsletter are written by members of TAPS unless otherwise noted and no guarantees are made as to the accuracy or validity of any information contained herein. Any opinions expressed in this newsletter are those of the individual authors and do not necessarily reflect those of the THE Atari Programmers' Society.

"Atari" is a trademark of Atari Inc. Any other names of products or companies are likewise trademarked and are used either with permission or as an informal method of referring to a product or company.

Articles were contributed this month by Nick Coleman, J.P. Fodor, Graham Thorley and P. Blake

Our apologies to Brenda Tupling, whose contribution reached us too late and to Sean Keely and Douglas Skrecky who submitted programs which were incorporated into an article for which there was no room.

Apologies also to ICD for having spelt their name more times wrong than right last month.

Permission is granted to any Atari Users' group to reprint articles contained in this newsletter on condition that credit is given to the author and to THE ATARI PROGRAMMERS' SOCIETY.

T H E A t a r i P r o g r a m m e r s' S o c i e t y
P.O. Box 6287 Stn. "A"
TORONTO ON M5W 1P7

THE Atari Programmers' Society (TAPS), formerly the Toronto Atari Programmers' Society, is an independent Atari computer users' group and is not affiliated with any commercial organization. Membership in TAPS is $25 a year.

WireTAPS EDITOR: Peter Blake (534-3486) SOFTWARE LIBRARY: Tim Glass (444-1957)

General Informaton:

CHAPTER	CONTACT	MEETS ON	MEETS AT
OSHAWA:	Brenda Tupling(1-668-6099) Howard Cruse(1- -)	Second Saturday, 10:30am	Whitby Public Library
SCARBOROUGH:	Graham Thorley(284-2369) Doug Barton(757-3042)	Second Tuesday, 7:30pm	Cedar Ridge Crafts Centre
TORONTO:	Sean Keeley(929-9330) Ian Davidson(848-0821)	First Thursday, 7:30pm	Norseman Community School

BBS: Toronto: The Sports Board (294-9730) Oshawa: The Red Baron (668-2078)

TAPS Affliated Organizations

PETERBOROUGH ATARI PROGRAMMERS (PAP): Dave Castell (742-8702)
Grant Johnston (743-6585)

PENINSULA ATARI COMPUTER ENTHUSIASTS (PACE): John Daskewich (892-6014 [Fonthill])
Alan Monroe (732-5027 [Welland])
BBS (357-2955 [Niagara Falls])

Action!, BASIC XL, BASIC XE, DOSXL, C65, MAC65, The Writer's Tool and OSS are registered trademarks of Optimized Systems Software. PaperClip is a registered trademark of Batteries Included. Print Shop is a trademark of Broderbund.

EDITORIAL

By Peter BLAKE

For the first time in a long time, WireTAPS has a graphics cover, and if TAPS' members keep coming up with good graphics, the practice will continue. So if you are into graphics, think about printing something for the WireTAPS' cover. If we get to be really affluent, we might even think of doing it in colour. Thanks for the cover goes to TAPS' member John Hammond, who has become something of an expert on using the Print Shop.

Contributions to WireTAPS are down this month. Please keep those articles coming. We all enjoy reading them. Unfortunately, because of the "unusual activity" in the Post Office around Christmas, the deadline for articles for the January 1986 issue of WireTAPS will be December 9th. As usual, exceptions will be made if applied for early, but do try to make that deadline. That issue of WireTAPS will be in the mail around December 17, well before the Christmas rush. That way, I will be able to have a few days' rest during the Christmas period.

Soon the shops will be full of new items which manufacturers hope that you will purchase for your computer for Christmas. By the time that you read this, the long-awaited Atariwriter Plus should be on your dealers' shelves. I hope to have a review of word processors for the Atari in next month's issue, or the one after that. The product which gets your Editor's award for **Innovation and Advancement of Personal Computers** is the Mouse-House by American Covers Inc. It's a cover for a mouse with fur and eyes and ears which turns your mouse into a real mouse-looking mouse. It seems to me that this is the kind of dynamic, positive thinking that the personal computer industry so desperately needs in order to humanize personal computers and to make them more accepted by the general non-computing public.

In my copious free time, I am slowly working on BASICASM version 2. Thus far I have eliminated some of the restrictions of version 1 and written the temporary work file out to a RAMDISK if there happens to be one available. This has improved the speed considerably again. I shall donating BASICASM Version 2 to the software library when it is fully operational and debugged.

I have given up for a moment working on the version in Action! It is proving to be more difficult than I had originally thought.

I guess that I should be taking advantage of the last issue of the year to look back over the year's activities and commenting upon them in a somewhat knowledgeable fashion. 1985: the year the Atari ST hit the market. It has been a water-shed year for computer manufacturers. A lot of companies and products did not make it to the end of the year. Atari did, and there were a lot of people who did not think that it would. The Amiga also came along. It is an impressive machine. In the long run however (this is a theory that I have long held), it is not the hardware that makes or breaks a computer: it is the software that is available to be run on it.

The fact is that there is still a long way to go and lots of software that can be written for the 6502-based ("8-bit") Ataris. Writing software for 6502-based machines is still just coming out of the middle ages. There is a lot that can be done in terms of writing software that is flexible, and "user friendly". By "user-friendly" I do not necessarily mean "menu-driven". I mean rather software that lets the user define, for example, how he or she wants "function keys" set. MEDIT, my favourite text editor uses <CNTRL 9> to forward space a page. The Writer's Tool, my favourite word processor uses <CNTRL F> to accomplish the same thing. If I were able to define what keys I wanted to press to scroll forward one screen, I would be able to set up all of the utilities I used to function in the same way. This would make my life a whole lot simpler. That is just one small example of what I mean by "user friendly".

Atari Corp. contacted us from California to inform us that they were sending us a copy of ST BASIC. With any luck at all, it should be here by the December meeting. TAPS' members who are ST owners should arrange to get a copy of it when it arrives.

Those TAPS' members who ordered BASIC from Irwin, please hold on. We are still trying to get the cartridges for you.

If you did not get last month's issue of WireTAPS, please let a member of the Executive know. We seem to have had another glitch with our membership database.

If you have written to us and not received a reply, please be patient. Our Post Office Box is emptied twice a week, then it takes a little longer for the correspondence to get to the person who deals with it. We will try to answer all the letters soon.

TAPS SCARBOROUGH

Well unfortunately the last meeting happened to be on the same night as the municipal elections (the previous month it was the Blue Jays first playoff game). That plus the rain seemed to keep

attendance down.

We started off by handing out new TAPS membership application forms plus a survey for completion. The new membership form is to allow us to get a better profile on the typical TAPS member.The survey will help the club serve your needs better. Tim Glass showed off the latest software disk and for those not in attendance there is a very good utility file which lets you design a title page for your programs and automatically writes the lines to append to your program.

Chris Thomson showed off his latest piece of software "Ultradisassembler" and Doug Barton demo'd two new games "Karateka" and "Koronis Rift". All of these programs looked to be excellant and we thank Chris and Doug for bringing them in.

We would like to welcome the following new member:

Ronald Weigand

NEXT MEETING

Tuesday Dec. 10th, 7.30 p.m. sharp at Cedar Ridge Craft Centre, 225 Confederation Dr. Scarborough.

MODEM NEWS

Last month we indicated that TAPS has its own message base on THE SPORTS BBS. All members of TAPS have been validated with the exception of those who would have to call long distance. Should anyone who is not validated wish to be, then please leave a message with the Sysop and he will look after you. Should there be enough membership support of the TAPS base we will consider setting up a larger section of the BBS for TAPS' members' private use with Tutorials, d/l's, etc.

So if you have a modem, give the board a call and leave a message even if it just says "hello".

THE SPORTS BBS (416) 294-9730 24 hours

CAN YOU HANDLE IT?

I have been writing articles for WireTAPS fairly regularily, lately based around the minutes of the Toronto meeting. However, this month I was so impressed with one of the programs on Disk Issue #31 that I thought I would devote some time to describing its capabilities instead.

This is not to say that I will omit any further mention of the meeting, but just not quite as much as usual.

Perhaps I had better do that first. Once again, one of TAPS great annual events in which all members actively participate was successfully completed by those in attendence. This is of course our latest survey, which again this year took everyone by surprise as, before they could escape, Tim had already distributed the questionnaires. As always, a tabulatiion of the results has been promised and with any luck, this time we'll actually get some feedback back.

The software library was successfully demonstrated, but not before the customary battle with the school's monitor which all but refuses to accept video input from an ATARI. Tim and Doug managed to subdue the beast, this time even with out my elastic band.

Joe Fodor, chaired the meeting and indicated several new concepts that he and the rest of the Toronto executive would like to introduce. These included door prizes (free disk of the month) at each meeting, telephone numbers to call when you need help with your ATARI (see the article elsewhere in this issue), a suggestion/problem box to be used for, you guessed it, problems and suggestions, and a request that each member remember to bring a friend to the next meeting.

Sean graciously demonstated something old, STAR RAIDERS, and something new, F-15 Strike Eagle. As usual Sean tried to face the group when speaking, which is unfortunate as although he appeared innately adept at avoiding enemy missles even when not watching the screen, the same cannot be said for his seat-of-the-pants flying, that eventually almost ended in a severe disruption of his F-15's ability to be recognized as an airplane.

Back to my original subject. "G:" from the October 1985 issue of ANALOG, written by Charles F. Johnson, was included on Disk #31 as GHANDLER.OBJ. It works as an AUTORUN.SYS file and is completely transparent to the user, which means you do not even know it's there. What it does, is establish another device that your computer can talk to. You are no doubt already familiar with D: for disk drive and P: for printer etc. The new device is G: for a Graphics handler. So what, you say? Well hold on to your hat and read further. Whenever you wish to list your programs to the printer including all graphics symbols, inverses and control codes, just type LIST "G:". That's it! No more having to load another program to transfer a copy of your program that you previously had to LIST not SAVE to the disk. Furthermore, LISTing to G: prints 114 single width characters to a line, which is the longest program line the ATARI allows; G2: prints each line as it would appear on the screen with single width and margin set at 2; G3: prints the same as G2: but in double width; G4: prints 60 double width characters to a line. Still holding onto your hat? GHANDLER also prints any Graphics mode 0, 7+ and 8 screens just as easily and in single or double height and single, double or triple width. Thus by including a function in any of your programs, such as a push of the START button, you could dump the present screen to the printer. These capabilities and others are more fully explained in ANALOG, a copy of which I suggest you acquire in order to get maximum benefit from this remarkable program.

Don't forget, its up to you to support your club, WireTAPS and the Software Library.

Same Place-Different Time, Sort of

Starting in December, the Toronto meeting at Norseman will have the doors opened at 7:00pm rather than 7:30pm. Athough the meeting still won't start until 7:30, you won't have to wait in the hall for the room to be unlocked. This will also facilitate those who wish to come early and use the extra time to talk to other members prior to the meeting.

BEWARE THE WORMS!!!

No this is not the latest horror movie released for the Christmas Season, though if it happens to you, it most certainly could be a horror story. I am of course presuming that accidentally formatting one of your favourite disks resulting in the unintentional loss of all the resident programs, would not necessarily be what you might consider fun, but rather, and more likely, a major catastrophe. Well, it seems that there are a few demented weirdos out there, whose very most favourite recreational activity, is to upload files to their local bulletin boards. These are no ordinary programs mind you. Their titles appear quite appealing, enough so that most bulletin board users would probably choose to download them. The unfortunate thing is that these programs' only purpose in life is to cause your disk drive to execute an untimely format of the resident disk, wiping out all your files. The moral here, is that if you download any programs from a bulletin board, transfer them to a blank disk before running them. This is about the only safe way to determine just what your latest acquisition is going to do, without running the risk of an unwelcome surprise. If you discover such a program, please leave a message with the sysop, so he can remove it before any other unsuspecting users suffer unnecessarily.

SAVE SOME MONEY, MAYBE

It appears that a number of our members are considering the purchase of BASIC XE and/or Action!, both from Optimized Systems Software Inc. It was requested that we investigate the possiblity of making a group purchase, with the aim of obtaining some form of discount. We will endeavour to solicit such an undertaking from some of the ATARI dealers around town, but we would also like to know how many would be interested. So far, the Toronto chapter would like 6 copies of ACTION! and 4 copies of BASIC XE. Although we do not have any idea yet what the reduction in cost might be, it is obvious that higher volume means lower prices. Please let somebody on the credits page know within the next few weeks, and we'll see what we can do for you. Once the number of copies and discount has been set, those still interested will be asked to provide funds in advance.

[This article is reproduced, with permission, from **PAGE SIX**, the newsletter of the Caltari Users' Group in Calgary. It was keyed in by Sue and Joe Fodor, whom we would like to thank very much. Joe has added a few sentences here and there to help you understand some concepts a little better.]

STANDARD ATARI DISK FORMAT

by Bruce SMITH, CALTARI Users' Group

AND IN THE BEGINNING

This is a re-write of the handout passed out at the September 5th general meeting [of the CALTARI Users' Group]. I have restructured it somewhat, plus filled some gaps and corrected a few errors. This is a better reference document than the handout, and so I would suggest you throw out the handout...

Later on in the article, you will see some 2-digit numbers prefixed by a "$" (e.g. $03). This is the ATARI standard method for defining a hexadecimal (base 16) number. The digits in the hexadecimal system are 0-9 and A-F (translating to 10-15 in decimal mode). By using hexadecimal numbers, it is possible to represent the numeric value of 1 byte in just 2 digits, making it easier to interpret it as individual bits (once you get used to the numbering system), plus it reduces the amount of space needed to display the number (2 locations instead of 3, since a byte can hold a decimal value from 0 to 255).

You will often find that 2-byte numbers and addresses on the ATARI are stored in low/high format (the value as we would understand it is 256*high+low). This is because that is the way the 6502 microprocessor chip (the heart of your ATARI) works with numbers and addresses. Many computers (including the human brain) work the opposite way (high/low), but not your trusty 6502.

And now, on with the show.

OVERVIEW

Standard Atari-formated disks (the term is DOS 2 disks) have 720 sectors, numbered from 1 to 720. These are laid out in 40 tracks (40 concentric rings on the diskette) of 18 sectors each. Enhanced density disks (1050 Dual density) have 40 tracks of 26 sectors each, yealding 1040 sectors in total. Getting back to DOS 2 format disks, of the 720 sectors available, 13 are reserved for special functions:

SECTOR	USAGE
1	initial boot sector
2	remainder of boot
3	garbage, but part of boot
360	VTOC
361-368	DIRECTORY
720	unused

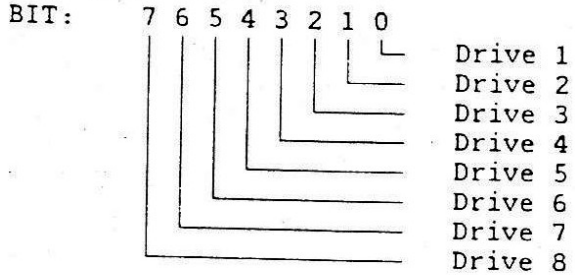
This leaves 707 sectors free for use as data storage.

The first 3 sectors are always 128 bytes long, while the remainder are either 128 bytes for single density, or 256 bytes for double density.

[SEE: SECTOR LAYOUT - DOS 2 FORMAT DISKS DOS BOOT - SECTOR 1 on page 5.

Some of the flag bytes deserve more explanation:

BYTE 10 - ACTIVE DRIVES FLAG - This is a set of bits that define which drive(s) are available for use by DOS.



Typically, bit 0 will be on (giving $01), meaning drive 1 is available, or bits 0 and 1 will be on (giving $03), meaning drives 1 and 2 are available.

BYTE 14 - DOS FLAG - This byte flags whether DOS has been written on the disk (i.e. DOS.SYS exists on the disk), and if so, what density it is: $00=no DOS, $01=single density DOS, $02=double density DOS

BYTE 17 - SECTOR LINK OFFSET - This contains the offset within a sector of the sector link information (described later). In single density, this will be $7D, and in double density will be $FD.

[SEE: DOS BOOT - SECTOR 2 on page 5.]

[SEE: DOS BOOT - SECTOR 3 on page 5.]

[SEE: VTOC - Sector 360 on page 5.]

VTOC Table of bits to sector numbers:

bit	7	6	5	4	3	2	1	0
byte 10	-	1	2	3	4	5	6	7
byte 11	8	9	10	11	12	13	14	15
.								
.								
byte 99	712	713	714	715	716	717	718	719

This sector [the VTOC] is used to keep track of disk sector usage. The most important part of the VTOC is the sector bit map.

A zero bit in the sector bit map means that the sector is in use. Note that sector 0 has a slot reserved for it, and sector 720 does not. There was a communication mixup between the hardware developers and the authors of DOS. The hardware recognizes sectors 1 to 720, but the developers thought it was to be 0 to 719 (DOS was written without having a real machine to work on.... I think Bill Wilkinson and friends did an admirable job, considering; BASIC REV. A was developed the same way).

The sector bit map is a contiguous string of 90 bytes, each containing 8 bits for a total of (90 X 8) 720. The 90 bytes of bit map start at VTOC byte 10 ($0A). Since sectors 1 to 3 are reserved, on an empty disk byte 10 should contain $0F. Then there will be a lot of $FF's until we get to byte 55 (which covers sectors 360 to 367) which will be $00, and byte 56 (which covers sectors 368 to 375) which will be $7F. The remaining bytes up to 99 will again be all $FF's.

The above structure makes calculating sector numbers/bit offsets fairly easy, since modulo 8 arithmetic can be used to convert back and forth, which is why Bill et al. did it that way, and did not fix the problem later on, when it was uncovered. It would have been too difficult, so they left it alone.

On a double density disk, only the first 128 bytes are used. The remainder of the sector is ignored.

[SEE: DIRECTORY - SECTORS 361 to 368 on page 6.

Sectors 361 to 368 were chosen for the Directory because they are in the middle of the disk and therefore have the minimum average seek time. There are 8 sectors, with 8 entries in each (hence the limit of 64 files per disk). Each file entry is 16 bytes long, and is laid out as follows:

FLAG VALUES: $80 = deleted
$40 = exists
$20 = locked
$01 = open for output

These bits can be merged together, allowing the following 5 combinations:

$80 = deleted
$60 = exists and is locked
$41 = exists and is open for output
$40 = exists
$00 = not used since disk formatted

Note that the directory only knows the first sector of a file. This explains why, when you delete a file, that you hear the disk reading sectors for awhile. In order to add the sectors from that file back onto the free sector list (in sector 360), DOS must find all the sectors, and so has to spin through the whole file.

Since 8 entries at 16 bytes each yields 128 bytes in total, the last half of the sector is ignored on a double density disk.

[SEE: USER DATA - SECTORS 4 TO 359 AND 369 TO 719 on page 6.]

SECTOR LAYOUT - DOS 2 FORMAT DISKS
DOS BOOT - SECTOR 1:

BYTE:	0	1	2	3	4	5	6 - 8	9
	xx	03	low | high		low | high			xx
	unused	no. sectrs to boot	boot load address		dos init address		jump to boot init code	max open files

BYTE:	10	11	12	13	14	15	16	17
Continued:	xx	xx	low | high		xx	low | high		xx
	active drives flag	unused	boot end address+1		DOS flag	starting sector of DOS.SYS		sector link offset

BYTE:	18	19	20	127
Continued:	low | high		108 bytes of data	
	DOS.SYS starting address		boot instructions	

DOS BOOT - SECTOR 2:

BYTE:	0	1	2	3	4	5	6	....	127
	128 bytes of data (more instructions)								

DOS BOOT - SECTOR 3:

BYTE:	0	1	2	3	4	5	6	....	127
	128 bytes of garbage....ignore it								

VTOC - Sector 360:

BYTE:	0	1	2	3	4	5-9	10-99	100-127
	flag	low | high		low | high		unused	VTOC	unused
	dir. type	max sector no -garbage, ignore		no. free sectors			sectors used bits	

DIRECTORY - SECTORS 361 to 368

BYTE:	0	1	2	3	4	6	12	13 - 15
	flag	low | high		low | high		name		extender
	see below	sector count		starting sector				

USER DATA - SECTORS 4 TO 359 AND 369 TO 719

	BYTE: 0 ... 124	125	126	127
Single Density	user data	file no./chain		count

	BYTE: 0 ... 252	253	254	255
Double Density	user data	file no./chain		count

FILE NUMBER / CHAIN

BYTE:	.....125 or 253.....								.....126 or 254.....							
BIT:	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	6 bit file number (0-63)								10 bit next sector number (0-1023)							

A data sector is used to contain the files data bytes. Each 128-byte data sector holds 125 bytes of data and 3 bytes of control information. The data bytes start in byte 0 and run contiguously up to (and including) byte 124. In byte 125, we have the sector byte count. It may range from 0 (no data) to 125 (full sector). Any data sector in a file may be a short sector (contain less than 125 bytes). The left 6 bits of byte 126 contain the file number (which is the file's relative position from the start of the directory). If they don't match (i.e. the sector chaining has gotten messed up), you will get an ERROR 164 (file number mismatch) when processing the file.

The right 2 bits of byte 126 and all 8 bits of byte 127 point to the next sector for the file. The 10-bit number contains the actual disk sector number of the next sector in the file sector chain. The value may be 0 to 719 ($2CF) and if it is 0, then there are no more sectors in the file chain.

By using a disk sector editor (there are several available in the Software Library), you can examine your disks in more detail, and perhaps gain a better understanding of this information.

[The original article goes on at this point to give a short description of DOS 2.5 format and DOS 3 format. Unfortunately, our copy of PAGE SIX has a badly printed page, and what is said is largely illegible. Ed.]

INTRODUCTION TO ASSEMBLY-LANGUAGE PROGRAMMING.

Part X - Comparing with the CP- instructions.

By Peter BLAKE

Thus far in this series, we have seen just about every instruction that is needed in order to write simple assembly-language programs. The only instructions that I have omitted or skirted around were the compare instructions: CMP, CPX and CPY. You may have noticed that frequently when the BASIC program said IF X > 79 the corresponding assembly-language statement said IF X = 80. You may have wondered why that was done. Today we shall find out why. But before we do that, a small review might be in order. This will save you trying to find back issues of WireTAPS to look up what is being talked about. Firstly, remember what the 6502 architecture looks like:

15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
	ACCUMULATOR	A
	"X" REGISTER	X
	"Y" REGISTER	Y
PROGRAM COUNTER		PC
	STACK POINTER	SP
	STATUS REGISTER	SR

The numbers at the top of the diagram represent bits. All of the registers but the Program Counter are 8 bits wide.

Next, here is what the Status Register (SR) looks like with an explanation of the flags which are used with compares:

S	V	n/u	B	D	I	Z	C

Where S = Sign Flag (also called Negative)
V = Overflow Flag

n/u means not used
B = Break Flag
D = Decimal Flag
I = IRQ (Interrupt) Flag
Z = Zero Flag
C = Carry Flag

The SIGN Flag is 1 (i.e. on) if the result of the previous instruction is negative. Remember that a negative number is one whose high-order bit is 1. As a one-byte number, this means any number higher than $7F is negative.

The ZERO Flag is set by certain instructions, when the number loaded or the result of the operation performed is zero.

The CARRY Bit should be thought of as the accumulator's ninth bit in addition and subtraction. When the result of an addition or subtraction is larger than the 8-bit accumulator will hold, the carry bit is turned on.

Remember also that the 6502 instruction set provides a set of eight "branch" instructions, a pair for each of four bits in the Status Register. These instructions are:

BCC Branch on Carry Clear (i.e. no carry)
BCS Branch on Carry Set (i.e. there is carry)
BEQ Branch on EQual to zero
BNE Branch on Not Equal to zero
BMI Branch on MInus (i.e. sign flag = 1)
BPL Branch on PLus (i.e. sign flag = 0)
BVC Branch on oVerflow Clear
BVS Branch on oVerflow Set

Now, the first rule to follow is: IF YOU CAN POSSIBLY USE AN EQUALITY COMPARISON (BEQ or BNE), USE IT. It is not always possible, and if there is any doubt, you will have to do it the long way. Let us start, however, by looking at the description of the CMP instruction.

CMP--CoMPare memory and accumulator.

The CMP instruction subtracts the contents of the memory location specified by the operand from the Accumulator. Neither the contents of the operand nor the contents of the Accumulator are changed, only the status flags are set.

If the contents of the accumulator and the contents of memory pointed to by the operand are equal, the Zero flag is set. That is the easy part.

If the result of the operation is greater than 127, the Sign (Negative) flag is set to one; otherwise the Sign flag is set to zero.

The Carry flag is set to one when the value stored in memory is less than or equal to the number stored in the accumulator. The Carry flag is set to zero when the number stored in memory is less than the contents of the accumulator.

The CMP instruction comes in the following addressing modes:

```
CODE      INSTRUCTION      comments
        . =   $3000        ;have to start somewhere
C966      CMP  #$66        ;IMMEDIATE ADDRESSING
C566      CMP  $66         ;ZERO PAGE ADDRESSING
CD1430    CMP  HISCOR      ;ABSOLUTE ADDRESSING
D566      CMP  $66,X       ;ZERO PAGE INDEXED
DD1530    CMP  FROM,X      ;INDEXED ADDRESSING
D91530    CMP  FROM,Y      ;INDEXED again
C166      CMP  ($66,X)     ;PRE-INDEXED INDIRECT
D166      CMP  ($66),Y     ;POST-INDEXED INDIRECT
        HISCOR .DBYTE 0
        FROM   .DBYTE 0
```

The **CPX** and **CPY** instructions work exactly like the CMP instruction except that instead of comparing the Accumulator, it is the X-Register or Y-Register that are being compared. Also, CPX and CPY have only three addressing modes available: IMMEDIATE (#$66), ABSOLUTE and ZERO-PAGE.

Although that explanation may seem on the surface to be quite straight forward, it is not. A lot more explanation and examples will be required before you will begin to have any fluency and confidence with these instructions.

One of the reasons for this is that one does not normally think in terms of Zero or Sign or Carry Flags. Normally, one thinks in terms of "greater than" or "greater than or equal to" or "less than" or "less than or equal to". This is in fact the way than most high-level languages like BASIC or Pascal or C or Action! work. Even some assembly languages have instructions like BEQ (Branch if EQual) BLE (Branch if Less than or Equal to). The 6502, alas, does not and this fact alone makes 6502 assembly-language programming very difficult.

While we were looking at adding, I mentioned that what you got depended on whether you considered that you were working with signed or unsigned numbers. Remember that a signed number treats the high-order bit as a sign ("1" = negative) and uses "two's complement" arithmetic. Again that means that $FF is -1, $FE is -2 etc.

Mostly, you will likely be using unsigned numbers, so you may chose to ignore what is being said about signed compares, just keep them in mind however.

First, let us try to translate some BASIC "IF" statements into their assembly-language equivalents:

```
BASIC: IF C=D THEN 100
ASSEMBLER: LDA C
           CMP D
           BEQ L100

BASIC: IF C<>D THEN 200
ASSEMBLER: LDA C
           CMP D
           BNE L200

BASIC: IF C<D THEN 300
ASSEMBLER: LDA C
           CMP D
           BCC L300

BASIC: IF C>D THEN 400
```

```
ASSEMBLER: LDA C
           CMP D
           BEQ NEXT ;if equal then not greater
           BCS L400
     NEXT  ... ...
```

```
BASIC: IF C<=D THEN 500
ASSEMBLER: LDA C
           CMP D
           BCC L500 ;less than so branch
           BEQ L500 ;equal so branch
     NOTTRU  =  .   ;C must be greater than D
```

The next set of examples is reference for those who might wish to use signed arithmetic. Only those cases which are different are listed:

```
BASIC: IF C<D THEN 300
ASSEMBLER: LDA C ;C & D are signed
           CMP D
           BMI L300
```

```
BASIC: IF C>D THEN 400
ASSEMBLER: LDA C ;C & D are signed
           CMP D
           BEQ NEXT ;if equal then not greater
           BPL L400
     NEXT  ... ...
```

```
BASIC: IF C<=D THEN 500
ASSEMBLER: LDA C ;C & D are signed
           CMP D
           BMI L500 ;less than so branch
           BEQ L500 ;equal so branch
     NOTTRU  =  .   ;C must be greater than D
```

Armed with this, we can try a small assembly-language program which illustrates branching. The one thing to note about the program is the way that the common instructions that are required to PRINT something have been put into the DSPLAY subroutine. This reduces the number of instructions that are needed to print something to six. Still more than BASIC but...

```
;-----------------------------------
; A SIMPLE GUESSING GAME TO ILLUSTRATE
;   6502 ASSEMBLY-LANGUAGE COMPARES
;  WRITTEN 1985-11-15 BY PETER BLAKE
;-----------------------------------
;
EOL    = $9B
OPEN   = $03
GETREC = $05
PUTREC = $09
CLOSE  = $0C
INPUT  = $04
OUTPUT = $08
EOF    = $88
;
ICHID  = $0340; Index to device
ICDNO  = $0341; device number
ICCOM  = $0342; I/O command
ICSTA  = $0343; STAtus
ICBAL  = $0344; LSB BUFFER ADDR
ICBAH  = $0345; MSB BUFFER ADDR
ICPTL  = $0346
ICPTH  = $0347
ICBLL  = $0348; LSB BUFFER LENG
ICBLH  = $0349; MSB BUFFER LENG
CIOV   = $E456
;
FRO    = $D4 ;6-byte Floating-Point Register 0
INBUFF = $F3 ;2-byte (MSB-LSB) pointer to input
;            ;buffer for F-P converstion
CIX    = $F2 ;1-byte offset in buffer
AFP    = $D800 ;Address of ATASCII to Floating-
;              ;Point conversion routine in ROM
FPI    = $D9D2 ;Address of Floating-Point to
;              ;Integer conversion routine
;
.      =   $4000 ;out of DOS's way
;
LP1    JSR INIT
       JSR GETRND ;Get a random number
LP2    JSR GETNUM
; if NUM = ANSWER then game over
       LDA NUM ;if random NUMber
       CMP ANSWER ;is equal to ANSWER
       BEQ RIGHT ;congratulate
       BCS TOOLOW ;else if ANSWER low
; else ANSWER must be too high so
TOOHI  JSR DSPLHI ;Display answer too high and
       JMP LP2 ;Loop and ask for another guess
;
TOOLOW JSR DSPLLO ;Display answer too low and
       JMP LP2 ;Loop and ask for another guess
;
RIGHT  JSR DSPLRT ;Congratulations!
       JMP LP1 ;and play another time
;
;-----------------------------------
; ROUTINE INIT - DISPLAY STARTING MSG
;-----------------------------------
INIT   LDX #0 ;use IOCB #0
       LDA #MHELLO ;LSB to be displayed
       STA ICBAL,X ;look familiar?
       LDA /MHELLO ;MSB to be displayed
       STA ICBAH,X ;into IOCB  #0
       JSR DSPLAY  ;common print instructions
       RTS ;return to calling routine
;===================================
;
;-----------------------------------
; ROUTIME GETRND--GET A RANDOM NUMBER
;-----------------------------------
GETRND LDA #99
       JSR RND ;get number between 0 and 99
       STA NUM ;and save
       INC NUM ;plus 1 to get between 1 and 100
       RTS ;return to caller
;===================================
;
;-----------------------------------
; ROUTINE GETNUM--ASK FOR & GET NUMBER
;-----------------------------------
GETNUM LDX #0 ;IOCB #0 as ever
       LDA #MENTER ;Print "ENTER YOUR
       STA ICBAL,X ;GUESS"
```

```
        LDA /MENTER
        STA ICBAH,X
        JSR DSPLAY
        LDA #BUFFER ;LSB of input buffer
        STA ICBAL,X ;tell IOCB where
        LDA /BUFFER ;MSB
        STA ICBAH,X ;you want answer
        STA ICBLH,X ;high-order zeroes
        LDA #100
        STA ICBLL,X ;any number would do
        LDA #GETREC ;command to read
        STA ICCOM,X ;
        JSR CIOV ;and let Atari OS do work
        LDA #BUFFER ;now let Floating-Point
        STA INBUFF ;routines convert number
        LDA /BUFFER ;into a binary number
        STA INBUFF+1 ;you can use
        LDA #0 ;see October WireTAPS
        STA CIX ;for details
        JSR AFP ;convert to Floating-point
        BCS GETNUM ;F-P did not like: try again
        JSR FPI ;convert F-P to integer
        BCS GETNUM ;F-P did not like: try again
        LDA FRO ;F-P puts answer in Floating-point
        STA ANSWER ;reg 1.  We put it in ANSWER
        RTS ;return to calling routine
;=====================================
;
;-------------------------------------
; ROUTINE DSPLHI--DISPLAY ANSWER TOO HI
;-------------------------------------
DSPLHI LDX #0 ;IOCB #0 once again
       LDA #MTOOHI
       STA ICBAL,X
       LDA /MTOOHI
       STA ICBAH,X
       JSR DSPLAY ;common PRINT instructions
       RTS
;=====================================
;
;-------------------------------------
; ROUTINE DSPLLO--SORRY ANSWER TOO LOW
;-------------------------------------
DSPLLO LDX #0
       LDA #MTOOLO
       STA ICBAL,X
       LDA /MTOOLO
       STA ICBAH,X
       JSR DSPLAY ;common PRINT instructions
       RTS
;=====================================
;
;-------------------------------------
; ROUTINE DSPLRT--CONGRATULATIONS RIGHT
;-------------------------------------
;
DSPLRT LDX #0
       LDA #MRIGHT
       STA ICBAL,X
       LDA /MRIGHT
       STA ICBAH,X
       JSR DSPLAY ;common PRINT instructions
       RTS
;=====================================
;
;-----------------------------------------
; ROUTINE DSPLAY--COMMON PRINT INSTRUCTIONS
;-----------------------------------------
DSPLAY LDX #0 ;insurance
       TXA ;fast way to get zero in accumulator
       STA ICBLH,X ;high-order zeroes
       LDA #100
       STA ICBLL,X ;any number would do
       LDA #PUTREC ;command to write
       STA ICCOM,X ;
       JSR CIOV ;Let OS do real work
       RTS ;return to calling routine
;=====================================
;
;-------------------------------------
;       WORKING-STORAGE SECTION
;-------------------------------------
;
MHELLO .TEXT 'HELLO, I AM THINKING OF A'
       .TEXT ' NUMBER      BETWEEN 1 AND 100.'
       .TEXT '  CAN YOU GUESS WHAT IT IS?  '
       .BYTE EOL ;EOL will end PUT
;
MENTER .TEXT 'PLEASE ENTER YOUR GUESS '
       .BYTE EOL ;ends line
MTOOHI .TEXT 'NO, THAT ANSWER IS TOO HIGH. '
       .BYTE EOL
MTOOLO .TEXT 'SORRY, THAT ANSWER IS TOO LOW.  '
       .BYTE EOL
MRIGHT .TEXT 'CONGRATULATIONS!!! YOU GUESSED RIGHT.
'
       .BYTE EOL
NUM    .BYTE 0 ;Random number
ANSWER .BYTE 0 ;Place to store answer
BUFFER .TEXT '                    '
;
       .IN D:GRAPHICS.ASM
;
       .END
```

Key the program in. It is not that long, and get it working. Now try some enhancements. Start off by asking if the person wants to play. If the answer is "NO" the first time, reformat the disk. See Nick's article on WORMS for how to do this. After every sucessful guess, ask if the player wants to play another game. If the answer is "NO" return to DOS. When doing this, split the GETNUM subroutine into another subroutine to avoid repetitive coding. Until next time when we will be looking at how to use the CIOV, season's greetings.

WHO YA GONNA CALL...?

By Nick COLEMAN

If your first guess was Ghostbusters, you'd be wrong. Recognizing TAPS' members' need for timely relief to their computing headaches, and further realizing that according to Murphy, these problems generally arise the day after the monthly meeting, where you might probably have got an answer, but

must now wait another 4 weeks, enter a new service, to be known, not as you might have second guessed as DIAL-A-PRAYER, but rather as DIAL-ANOTHER-TAPS-MEMBER-WHO-MAY-KNOW-MORE-THAN-YOU-AND-IS-WILLING-TO-HELP-SOLVE-YOUR-PROBLEMS-OVER-THE-PHONE. This new service may be referred to by its catchy acronym, "DATMWMKMTYAIWTHSYPOTP", especially if you are into tongue twisters and are slightly masochistic.

Some of our members that are more experienced in one thing or another, have agreed to provide help to our less experienced members over the phone. For those not already on the phone list, if you have any sort of expertise at anything, BASIC, DOS, Modems/Terminal programs, Logo, FORTH, Pascal, Action!, Assembly Language, Word processing, Databases, Spreadsheets, Adventure Games, Simulations, Touch tablets, Light pens, or even PACMAN, and would not mind helping other members who might have problems from time-to-time, please call one of the Executive and give us your name, number, and area of expertise.

Here in fact is a short list of those who have already volunteered:

Victor Brooks [Synfile,Syncalc] 625-6570 (Mississauga)
Sean Keeley [Action!,Logo] 929-3390 (Toronto)
Norm Hyde [General Info] 741-1468 (Toronto)

WANT ADS

[Want Ads may be placed free of charge by any TAPS' member in good standing.]

FOR SALE: Atari 400 Computer complete with BASIC cartridge. This is just the thing for families with small children, or backup in case your computer fails. Asking $50. Call G. Kay (416) 723-9017 (Oshawa)

[Your Editor has the awful feeling that someone gave him a WANTAD that he has misplaced. If that was you, please accept my apologies. Call me again, and I'll try again next month.]

DON'T THROW OUT YOUR MEMBERSHIP CARDS

It seems that some new members have not yet received their membership cards. We have checked our files and found that people who joined in September, should have received their cards by now. If you haven't, chances are that you accidently threw it out with the envelope. Anyone who has still not received their new or renewal card with the December issue of WireTAPS, is asked to inform Graham Thorley at 284-2369. But please remember to check your envelope thoroughly first.

4-MEGABYTE RAM FOR 520ST?

Well not yet, but soon. If you had attended the recent Computer Show, you would have seen a new LOTUS 1-2-3 look-a-like known as the VIP Professional already running on a 1-megabyte Atari 520ST. The 4 megs will likely be available next year through third party vendors. Currently, the 1-meg upgrade should be installed by an electronics technician, as it is extremely delicate work and one slip could burn out your new $1000 computer. It will take even an experienced technician about 3 1/2 hours to install the $10 or so worth of parts. Rumour has it that a board is being developed that will greatly reduce the risk and technician's time as only a few solder connections will be required.

Not yet available, but nevertheless in existence, is a 30-megabyte hard disk drive capable of loading 64k in about one second.

BMB Compuscience used their Imaginet Network to link an Atari 520ST, an IBM-PC and a COMPAQ so they could all run The Manager database (similar to dBase III) and access each others files at the same time.

There were quite a few excellent new programs for the 520ST demonstrated that will be on the shelves before December. These and others to be available soon comprised a 6-page list of new products for the 520. It looks like this machine is certainly here to stay and will be well supported by the software industry.

[The following is the text of an official press release issued Oct. 11 1985 by Sam Tramiel, president of Atari Corp., concerning the recent agreement between DRI and Apple.]

SUNNYVALE, CALIFORNIA -- In an agreement between Digital Research Inc. and Apple Computer, Digital Research has agreed to make certain superficial changes to three of its application programs: "GEM Desktop", "GEM Paint", and "GEM Draw". Contrary to reports, the agreement does not require changes to GEM or to Atari Corp.'s TOS operating system.

DRI, as part of its normal development process, has prepared enhancements to its GEM application products which further set it apart from the Apple Macintosh. Atari is reviewing these changes. Atari promises to its software developers and customers that Atari will make no changes that will reduce the capabilities of the Atari ST system and software. Further, Atari promises that any enhancements will maintain compatabiliy with GEM applications software of the market and in development.

Digital Research's GEM and the applications software which uses its capabilities will continue to provide the most powerful and easy-to-use computer interface on the market. The Atari 520ST continues to the the most powerful computer delivering these feature and will continue to do so at an affordable price.